ルータ上のバッファ管理問題に対するオンラインアルゴリズム

AUTHOR(S):
宮崎, 修一

CITATION:
宮崎, 修一. ルータ上のバッファ管理問題に対するオンラインアルゴリズム. 電子情報通信学会誌 2006, 89(4): 299-303

ISSUE DATE:
2006-04

URL:
http://hdl.handle.net/2433/227135

ルータ上のバッファ管理問題に対するオンラインアルゴリズム

Online Algorithms for Buffer Management Problems

宮崎修一

Abstract

ネットワーク上でふくそうが起った際に，ルータは到着するパケットをすべて処理しきれない場合がある．このとき，パケットの取捨選択やバッファの管理をいかに行うかが，QoS (Quality of Service) 保証においては重要となる．近年，この問題をオンライン問題として定式化し，オンラインアルゴリズムの競合比解析を行う研究が盛んに行われている．本稿ではこれらの結果を紹介する．

キーワード：バッファ管理問題，QoS保証，オンラインアルゴリズム，競合比解析

1. はじめに

インターネットにおけるパケットルーチングでは，ルータは入力ポートから受理したパケットのあて先アドレスを見，ルーチングテーブルを参照し，該当する出力ポートへと転送する．パケットの到着速度がルータの転送処理能力を超えれば，処理しきれないパケットはルータに受理されず，パケット損が発生する．パケット損を緩和させるため，通常，ルータにはバッファを設け，処理できないパケットをバッファ内で待たせておくという処置を施している．バッファ管理の最も単純なアルゴリズムは，バッファに空きがある限りパケットを受理するというものである．しかし近年は，例えばパケットに優先度（価値）を持たせ，価値の高いパケットは損失率を抑えるといった，いわゆるQoS (Quality of Service) 保証の考え方[1]が登場してきた．このような背景では，上記のような貪欲アルゴリズムは意味をなさず，パケットの価値に応じたバッファ管理を行わなければならない．従来，例えば，パケットの到着に確率分布を仮定しパケット損失率を解析するといった，いわゆる待ち行列理論により，アルゴリズムの性能が評価されていたが，近年このような問題をオンライン問題として取り扱う研究が登場した．

オンライン問題[2]とは，例えば株の売買や計算機内でのキャッシュメモリの管理を行うページングといった，時間の経過とともに情報が与えられ，将来の入力が分からない状況で各時点での判断を下す問題である．バッファ管理問題においては，パケットが到着した際，それを受理するのか，それとも将来到着するであろうより価値の高いパケットのために破棄するのかが，本質的な問題となる．アルゴリズムの性能は，競合比解析により行われる．Aをオンラインアルゴリズムとし，OPTを（将来も含む）すべての入力を知った上で最適な判断を下すアルゴリズムとする．入力σに対してAとOPTの得る価値（送ったパケットの価値の総和）をそれぞれ$A(\sigma)$，$OPT(\sigma)$とする．（すなわち，どのように頑張っても，入力列σからは$OPT(\sigma)$の価値しか得られない．）任意の入力σに対して$\frac{OPT(\sigma)}{A(\sigma)} \leq c$ならば，Aの競合比は$c$であるという．すなわち競合比は1に近いほど良い．計算量理論における最悪ケースの解析はしばしば現実とのギャップが問題視されるが，本問題におけるパケットのふくそうをルータへのDoS (Denial of Service) 攻撃ととらえた場合，ルータのアルゴリズムを熟知している攻撃者からのアタックをいかに緩和するかという現実的な問題と考えることができる．

本問題に対するモデルは多数考案されているが，本稿ではそれらの研究結果の概要を紹介する．

宮崎修一　正員　京都大学学術情報メディアセンター
E-mail　shuichi@media.kyoto-u.ac.jp
Shuichi MIYAZAKI, Member (Academic Center for Computing and Media Studies, Kyoto University, Kyoto-shi, 606-8501 Japan).
電子情報通信学会誌　Vol.89 No.4 pp.299-303 2006年4月

2. 単一バッファモデル

単一バッファモデルは最も単純であり，ルータ中の一つのバッファにのみ着目している．問題の入力は，「イベント」の列として与えられる．「到着イベント」ではパケットが1個到着し，「送信イベント」ではバッファからパケットが1個送信される．バッファの容量は B であり，パケットを同時に B 個までしかためておくことができない．アルゴリズムは，各到着イベントの後に，そのパケットを受理するか拒否するかを決定する．各パケットはそれぞれ価値を持ち，目標は送信されたパケットの価値の総和を最大化することである．以下では，バッファの能力によりモデルを三つのタイプに分類する．

2.1 Non-preemptive FIFO キューモデル

Non-preemptive FIFO キューモデルは本研究の先駆けとなった最も単純なモデルである[3]．送信イベントでは，キューの先頭にあるパケットが送信される．また，一度受理したパケットを途中で破棄することはできない．文献（3）では，各パケットが1または α (≥ 1) の価値を持つ2値の場合について様々なオンラインアルゴリズムを与え，それらの競合比を解析している．簡単のため，到着するパケットをバッファに空きがある限り受理するという，いわゆる貪欲アルゴリズムを考えてみよう．このアルゴリズムは，送るパケットの「個数」を最大化するため，OPT の送る個数以上は必ず送ることができる．OPT の送信したパケットがすべて価値 α であり，貪欲アルゴリズムの送信したパケットがすべて価値1だったとしても，OPT の $1/\alpha$ 以上の価値は得ているため，競合比は α 以下であるといえる．逆に，以下のような入力を考えてみる．最初の送信イベントの前に，価値1のパケットが到着するイベントが B 回起り，その後価値 α のパケットが到着するイベントが B 回起る．貪欲アルゴリズムは価値1のパケットをすべて受理し，価値 α のパケットを全く受理しない．その後到着イベントの起らないまま送信イベントが B 回起り，入力は終了する．貪欲アルゴリズムの得る価値は B である．一方 OPT は最初の B 個を見送り，価値 α のパケットをすべて受理するため，得る価値は αB である．したがって貪欲アルゴリズムはこの入力に対して都合が悪く，競合比がちょうど α だということが分かる．

文献（3）で提案されている中で最も性能の良いアルゴリズムは Dynamic Flexible Partition Policy（以下 DFP）で，その競合比の上限は2である．DFP は，価値 α のパケットはバッファに空きがある限り受理するが，価値1のパケットが到着した場合，現在のバッファ中の価値 α のパケット数と価値1のパケット数がある条件を満たせば受理する．（価値1のパケットを余り多く取り過ぎないように配慮している．）また，同じ文献（3）で，この問題の競合比の下限（すなわち，どんなアルゴリズムでもそれより良い競合比を持ち得ないこと）として，$2-\frac{1}{\alpha}$ が示されている．後に Andelman らは，DFP における価値1のパケットの受理条件を改良したアルゴリズムを提案し，その競合比 $2-\frac{1}{\alpha}$ を示した[4]．

自然な拡張として，パケットの持つ価値を1以上 α 以下の任意の値とする「多値モデル」がある．Andelman らはこのモデルに対して二つのアルゴリズムを提案し，どちらも競合比が $e\ln(\alpha)$ 以下であることを示した[4]．また，このモデルにおける競合比の下限が $1+\ln(\alpha)$ であることも示し，定数係数の範囲内で厳密な上下限を示している[4]．Andelman らは更に，上記のアルゴリズムを改良し，競合比 $\ln(\alpha)+2+O\left(\frac{\ln^2(\alpha)}{B}\right)$ を得ている[5]．これらの結果を表1にまとめる．

表1　Non-preemptive FIFO キューモデルに対する競合比の上下限

	上限	下限
2値	$2-\frac{1}{\alpha}$ [4]	$2-\frac{1}{\alpha}$ [3]
多値	$\ln(\alpha)+2+O\left(\frac{\ln^2(\alpha)}{B}\right)$ [5]	$\ln(\alpha)+1$ [4]

2.2 Preemptive FIFO キューモデル

Non-preemptive モデルでは，一度受理したパケットは最終的に送信される．これに対して，Preemptive FIFO キューモデルでは，一度受理してバッファ内にためられているパケットを途中で破棄すること（これを「プリエンプション」という）ができる．最適なオフラインアルゴリズムはプリエンプションを行わない（もし行うなら，初めからそのパケットを受理しなければよい）ので，プリエンプションを許すモデルはより強力である．現在最良の上下限を表2に示す．2値モデルに対しては，現在最良の競合比の上限は 1.304 [6]，下限は 1.281 [7,8] であり，上下限は非常に近い．一方，多値の場合は，貪欲アルゴリズム（バッファが満杯のときにパケットが到着したら，$B+1$ 個の中で最も価値の低いパケットを破棄するアルゴリズム）が競合比4を達成することが示され[9]，次いでその上限が $2-\frac{2}{\alpha+1}$ であることが示された[8]．文献（10）では，プリエンプションを貪欲に行うのでなく，既にキューに入っているパケットは多少価値が低くても残してやるアルゴリズム（β-Preemptive Greedy Algorithm：β は上記の「多少」の度合いを表すパラメータ）を提案し，初めて2を切る競合比 1.983 を

表2　Preemptive FIFOキューモデルに対する競合比の上下限

	上限	下限
2値	1.304 [6]	1.281 [7,8]
多値	1.75 [11]	1.419 [10]

示した．更に，プリエンプションの条件を改良することにより，競合比は1.75へと改良されている[11]．下限については，1.25が文献（23）で示され，その後1.281[8]，$\sqrt{2}$[4]，1.419[10]と改良された．

ここでは，比較的簡単な2値モデルの1.281下限の証明[8]を紹介する．簡単のため，送信イベントの起る時刻を整数（1,2,3,…）とする．以下の入力列を考えてみる．時刻1以前に，価値1のパケットが連続してB個到着する．その後，整数i（$i \geq 1$）に対して時刻$i+0.5$に価値αのパケットが1個ずつ到着する（αは後で決める）．つまり，この状況ではパケットを1個も落とすことなくすべて受理できることになる．Aを任意のオンラインアルゴリズムとし，この入力に対してAが初めて価値αのパケットを送る時刻を$t+1$とする．つまり，Aは時刻1以前に到着したB個のパケットのうち，t個を送り，残りは破棄している．ここまでは同じで，これ以降が異なる二つの入力を考える．第1の入力は，到着イベントはここでストップし，後は送信イベントのみとなる．Aは価値1のパケットをt個しか送っていない．また，時刻$t+1$でストップしたので，価値αのパケットはt個しか到着していない．したがってAの得る価値は高々$t+\alpha t$である．一方，最適なオフラインアルゴリズム（OPT）は到着したパケットをすべて送信することができるから，得る価値は$B+\alpha t$である．次に第2の入力を考える．時刻$t+1$において価値αのパケットを送信するためには，Aは時刻$t+1$以前に価値αのパケットをキューの先頭に持ってきておかなければならない．第2の入力では，その直後で時刻$t+1$以前に，価値αのパケットが連続してB個到着して，その後は送信イベントしか起らない．ここから後，Aはどんなに頑張っても価値αBしか得られないので，Aの得る価値は高々$t+\alpha B$である．OPTは価値1のパケットを全く受理しないことにより，価値αのパケットをすべて送信することができるから，得る価値は$\alpha(B+t-1)$以上となる．第1の場合と第2の場合を考慮すると，Aの競合比の下限は

$$\max\left\{\frac{B+\alpha t}{t+\alpha t},\ \frac{\alpha(B+t-1)}{t+\alpha B}\right\} \qquad (1)$$

となる．Aはαの値に応じて設計されたものなので，そのαに対して式（1）の値が最小になるようにtが選ばれている場合が，我々（下限を証明する側）にとって最も都合が悪い．そのようにtが選ばれるときの式（1）の値が最大となるようにαを選ぶと，$\alpha \simeq 4.01$となり，そのときの式（1）の値が1.281となる．上記は1とαの2値で2通りのシナリオを作っているが，文献（4）や文献（10）では多値で複数通りのシナリオを用意することにより，より良い下限を得ている．

2.3　Bounded Delayモデル

バッファがFIFOキューでないモデルとして，Bounded Delayモデルが考案されている．このモデルのバッファはFIFOキューではなく，パケットの到着順序にかかわりなく任意の順序で送信することが可能である．したがって，アルゴリズムは送信イベントの際にどのパケットを送るかも決定する．ただし，各パケットが有効期限を持っており，期限内に送信することができなければそのパケットは消失してしまう．このモデルに対する現在知られている最良の競合比を表3に示す．すべて多値モデルである．s-uniformはすべてのパケットの有効期限がちょうどsのモデルであり，s-boundedは，有効期限はs以下で，パケットによって異なってよいモデルである．

3.　マルチバッファモデル

よりルータに近いモデルとして，入力ポート，更には出力ポートを複数持つモデルが考案されている．ここでは代表的な三つのモデルを紹介する．

3.1　マルチキューモデル

このモデル[16]～[19]では，ルータは2.1，2.2で見たようなFIFOキューをm（≥ 2）本持っている．到着イベントでは，パケットがいずれかのキューに到着し，アルゴリズムはそのキューへの受理／非受理／プリエンプションを決定する．送信イベントでは，アルゴリズムはキューを一つ選択し，そのキューの先頭にあるパケットを1個送信する．このモデルでは，パケットの持つ価値が1種類であるとしても，問題は自明ではない．Azarらは，単一キューの場合の競合比cのアルゴリズムを利用して，本節のマルチキューモデルに対する競合比$2c$

表3　Bounded Delayモデルに対する競合比の上下限

	上限	下限
2-uniform	1.377[12]	1.377[12]
2-bounded	1.618[8]	1.618[4],[13],[14]
s-uniform	1.618[15]	1
s-bounded	1.939[12]	1.618[4],[13],[14]

表4　マルチキューモデルに対する競合比の上下限

		上限	下限
プリエンプションあり	1値	1.582[19]	1.582[18]
	2値	2.60[6],[16]	1.582[18]
	多値	3[20]	1.582[18]
プリエンプションなし	1値	1.582[19]	1.582[18]
	2値	$4-2/\alpha$[4],[16]	1.582[18]
	多値	$2\ln(\alpha)$[5],[16]	1.582[18]

以下のアルゴリズムを構成できることを示した[16]．また，Azarらは，パケットの価値の大小関係のみによって振る舞いを決定するアルゴリズムは，競合比解析において価値が0と1のパケットのみを考えれば十分であること（0/1原理）を示した[20]．現在の結果を表4にまとめる．2値の場合はパケットの価値は1とα，多値の場合は1以上α以下である．

3.2 CIOQモデル

CIOQ（Combined Input and Output Queued）モデルでは，N入力N出力のルータを考える．入力ポートと出力ポートはそれぞれ容量の制限されたキューを持つ．1ラウンドは，入力パケットの到着，スケジューリング，パケットの送信の3フェーズに分けられる．到着イベントはこれまで見てきたものと同様であり，アルゴリズムはパケットの受理／非受理／プリエンプションを決定する．スケジューリングフェーズでは，入力ポートと出力ポートの間のマッチングを作り，そのマッチングに沿って各入力キューの先頭のパケットが出力キューへと運ばれる．内部スピードアップというパラメータ（S）があり，1フェーズの中でマッチングがS回繰り返される．送信フェーズでは，各出力キューから先頭のパケットが送信される．（すなわち，一度に最大N個のパケットを送信することができる．）CIOQモデルにおける結果を表5に示す．ここでは，バッファはプリエンプションを許すFIFOキューである．

3.3 共有メモリモデル

このモデルでは，ルータはN本のFIFOキューを持つ．パケットの到着イベントについては，マルチキューモデルと同様であるが，送信イベントでは，各キューの先頭のパケットが送信される．また，保持できるパケット数について，各キューに対する制限はないが，すべてのキュー合計でM個に制限されている．このモデルに対する1値の場合，すなわち，アルゴリズムの性能が送ったパケット数で評価される場合の競合比の上下限を表6に示す．

表5　CIOQモデルに対する競合比の上下限

	上限	下限
1値	3[21]	1
多値	9.47[22]	1

表6　共有メモリモデルに対する競合比の上下限

	上限	下限
プリエンプションあり	2[23]	4/3[23]
プリエンプションなし	$\ln(N)+2$[24]	$\Omega\left(\frac{\log N}{\log\log N}\right)$[24]

4．おわりに

本稿では，バッファ管理問題に対するオンラインアルゴリズムの最近の研究動向について紹介した．誌面の都合上紹介できなかったが，紹介したモデルに対して，乱数を用いたアルゴリズムも多数考案されている．また近年では，複数のルータをネットワーク上に配置したモデル上でのバッファ管理やルーチングアルゴリズムの競合比解析という，より複雑な問題も登場している．

謝辞　本稿に目を通して頂き貴重な御意見を下さった，京大学術情報メディアセンター岡部寿男教授及び，東工大学術国際情報センター伊東利哉教授に感謝致します．また，本テーマに関する論文調査に御協力下さった京大大学院情報学研究科の修士課程学生である小林浩二氏に感謝致します．

文　献

(1) S. Blake, D.L. Black, M. A. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service," RFC 2475.
(2) A. Borodin and R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998.
(3) W. Aiello, Y. Mansour, S. Rajagopolan, and A. Rosen, "Competitive queue policies for differentiated services," IEEE INFOCOM, pp.431-440, 2000.
(4) N. Andelman, Y. Mansour, and A. Zhu, "Competitive queueing policies for QoS switches," Proc. SODA 2003, pp.761-770, 2003.
(5) N. Andelman and Y. Mansour, "Competitive management of non-preemptive queues with multiple values," Proc. DISC 2003, pp.166-180, 2003.
(6) Z. Lotker and B. Patt-Shamir, "Nearly optimal FIFO buffer management for DiffServ," Proc. PODC 2002, pp.134-142, 2002.
(7) M. Sviridenko, "A lower bound for on-line algorithms in the FIFO model," unpublished manuscript, 2001.
(8) A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko, "Buffer overflow management in QoS switches," SIAM J. Comput., vol.33, pp. 563-583, 2004.
(9) Y. Mansour, B. Patt-Shamrr, and O. Lapid, "Optimal smoothing schedules for real-time streams," Proc. PODC 2000, pp.21-29, 2000.
(10) A. Kesselman, Y. Mansour, and R. Stee, "Improved competitive guarantees for QoS buffering," Proc. ESA 2003, pp.361-372, 2003.
(11) N. Bansal, L. Fleischer, T. Kimbrel, M. Mahdian, B. Schieber, and M. Sviridenko, "Further improvements in competitive guarantees for QoS buffering," Proc. ICALP 2004, pp.196-207, 2004.
(12) M. Chrobak, W. Jawor, J. Sgall, and T. Tichý, "Improved online algorithms for buffer management in QoS switches," Proc. ESA 2004, pp.204-215, 2004.
(13) F. Chin and S. Fung, "Online scheduling for partial job values: Does timesharing or randomization help?" Algorithmica, vol.37, pp,149-164, 2003.
(14) B. Hajek, "On the competitiveness of on-line scheduling of unit-length packets with hard dead-lines in slotted time," Proc. CISS 2001, pp.434-439, 2001.
(15) F. Li, J. Sethuraman, and C. Stein, "An optimal on-line algorithm for packet scheduling with agreeable deadlines," Proc. SODA 2005, pp.801-802, 2005.
(16) Y. Azar and Y. Richter, "Management of multiqueue switches

in QoS networks," Proc. STOC 2003, pp.82-89, 2003.

(17) T. Itoh and T. Nagumo, "Improved lower bounds for competitive ratio of multi-queue switches in QoS networks," IEICE Trans. Fundamentals, vol.E88-A, no.5, pp.1155-1165, May 2005.

(18) S. Albers and M. Schmidt, "On the performance of greedy algorithms in packet buffering," Proc. STOC 2004, pp.35-44, 2004.

(19) Y. Azar and A. Litichevskey, "Maximizing through-put in multi-queue switches," Proc. ESA 2004, pp.53-64, 2004.

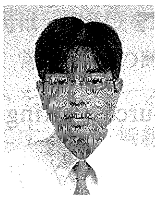
(20) Y. Azar and Y. Richter, "The zero-one principle for switching networks," Proc. STOC 2004, pp.64-71, 2004.

(21) A. Kesselman and A. Rosen, "Scheduling policies for CIOQ switches," Proc. SPAA 2003, pp.353-362, 2003.

(22) Y. Azar and Y. Richter, "An improved algorithm for CIOQ switches," Proc. ESA 2004, pp.65-76, 2004.

(23) E. Hahne, A. Kesselman, and Y. Mansour, "Competitive buffer management for shared-memory switches," Proc. SPAA 2001, pp.53-58, 2001.

(24) A. Kesselman and Y. Mansour, "Harmonic buffer management policy for shared memory switches," Theor. Comput. Sci., vol.324, no.2-3, pp.161-182, 2004.



宮崎　修一（正員）
平 5 九大・工・情報卒．平 7 同大学院修士課程了．平 10 同大学院博士課程了．同年京大・情報学研究科・助手．平 14 同学術情報メディアセンター助教授．博士（工学）．アルゴリズム，計算量理論の研究に従事．